Estructuras de Datos 1 - ST0245
Supletorio del Examen Parcial 2

Nombre:.....
Departamento de Informática y Sistemas
Universidad EAFIT

Octubre 30 de 2017

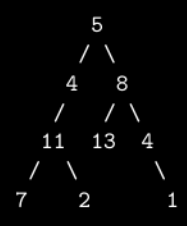
Criterios de calificación

- Selección múltiple con única respuesta
  - Respuesta correcta: 100 %
  - Respuesta incorrecta: 0 %
- Completar código
  - Respuesta correcta 100 %
  - Respuesta incorrecta o vacía 0 %

NOTAS IMPORTANTES:
- Responda en la hoja de PREGUNTAS
- Marque la hoja de PREGUNTAS

1. Árboles binarios 30 %

Definimos el camino desde la raíz hasta una hoja en un árbol binario como una sequencia de nodos empezando en el nodo raíz y bajando hasta una hoja. Una hoja es un nodo que no tiene hijos. Decimos que un árbol vacío no contiene caminos desde la raíz hasta una hoja. Como un ejemplo, el siguiente árbol tiene 4 caminos desde la raíz hasta una hoja:



Caminos desde la raíz hasta una hoja:

- camino 1: 5 4 11 7
- camino 2: 5 4 11 2
- camino 3: 5 8 13
- camino 4: 5 8 4 1

```
1 public class Nodo {
2   public int dato;
3   public Nodo izq;
4   public Nodo der;
5 }
```

Para este problema nos interesan las sumas de los elementos en los nodos de esos caminos, por ejemplo, la suma del camino $[5, 4, 11, 7]$ es 27. Dada la raíz de un árbol binario `Nodo a` y un entero `int suma`, **decir si existe un camino desde la raíz hasta una hoja tal que al sumar los valores de los nodos de ese camino la suma sea igual al parámetro** `suma`. Retorne falso si no se puede encontrar un camino con esa condición. Utilice la definición de `Nodo` del punto 1.

Desafortunadamente, al código que hizo Dayla le faltan unas líneas y Kefo está de vacaciones.

```
1 boolean sumaElCamino(Nodo a, int suma) {
2  if (a == null)
3    return __________;
4  else
5    return sumaElCamino(_______, _________)
6        || sumaElCamino(_______, _________); }
```

(10 %) Complete el espacio de la línea 03
_____

(10 %) Complete los espacios de la línea 05
__________, __________

(10 %) Complete los espacios de la línea 06
__________, __________

2. Colas 20 %

(10 %) ¿Cuál es la salida del siguiente algoritmo? El método `q.poll()` retorna y remueve el primer elemento de la cola `q`.

```
public void misterio(Queue<Integer> q){
  q.add(1);
  q.add(3);
  q.add(5);
  q.add(7);
  q.add(2);
  q.add(4);
  System.out.println(q.poll());
  while(q.size() > 0)
    System.out.print(", " + q.poll());
}
```

A. 4, 2, 7, 5, 3, 1
B. 1, 3, 5, 7, 2, 4
C. 1, 3, 5, 7, 4, 2
D. 1, 3, 5, 7, 4, 2

(10 %) Asuma que al algoritmo anterior se le ingresan n elementos. ¿Cuál es su complejida asintótica para el peor de los casos? El método `q.poll()` retorna y remueve el primer elemento de la cola `q`.

A. $O(n)$
B. $O(n^2)$
C. $O(n \log n)$
D. $O(2^n)$

3. Colas 20 %

(10 %) ¿Cuál es la complejidad asontótica en el peor de los casos de obtener el elemento en la posición i, $0 \leq i < n$ en una cola de n elementos?

A. $O(n^2)$
B. $O(n)$
C. $O(1)$
D. $O(\log n)$

(10 %) Se tiene el arreglo `int[] a = {1, 2, 3, 4, 5, 6, 7}` y la cola q vacía.

```
public void m(Queue<Integer> q, int[] a){
  for(int i = 6; i >= 0; --i){
    q.add(a[i] + 1);
  }
  while(q.size() > 0){
    System.out.println(q.poll());
  }
}
```

(10 %) ¿Cuál es la salida del anterior programa?

A. 7, 6, 5, 4, 3, 2, 1
B. 8, 7, 6, 5, 4, 3, 2
C. 1, 2, 3, 4, 5, 6, 7
D. 2, 3, 4, 5, 6, 7, 8

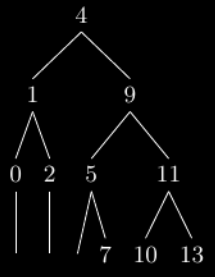
4. Árbol de busqueda 20 %

Los siguientes algoritmos son el recorrido en in orden y pos orden. En la vida real, estos recorridos son de utilidad para hacer procesamiento del lenguaje natural.

```
void InOrden(Node node){
 if (node != null){
   InOrden(node.left);
   System.out.println(node.data);
   InOrden(node.right);
} }

void PosOrden(Node node){
 if (node != null){
   PosOrden(node.left);
   PosOrden(node.right);
   System.out.println(node.data);
} }
```

Considere el recorrido in orden y pos orden de un **Arbol binario de busqueda** con los siguientes elementos 4, 9, 1, 5, 7, 11, 13, 2, 0, 10. Ahora, la impresión del recorrido in-orden nos devolverá los elementos $A_1, A_2, ..., A_i, ..., A_{10}$, y, la impresión del recorrido pos-orden devolverá $B_1, B_2, ..., B_i, ..., B_{10}$.



(10 %) ¿Cuál es la impresión pos-orden?

A. 0, 2, 1, 7, 5, 10, 13, 11, 9, 4
B. 0, 1, 2, 4, 5, 7, 9, 10, 11, 13
C. 0, 2, 1, 7, 10, 5, 13, 11, 9, 4
D. 4, 1, 0, 2, 9, 5, 7, 11, 13, 10

(10 %) Suponga que hacemos la comparación de A_i y B_i, ¿cuál es la cantidad de elementos para los cuales A_i no es igual a B_i?

A. 7
B. 2
C. 5
D. 8

5. Pilas 10 %

¿Cuál es la salida del siguiente programa?

```
public void mis(Stack<Integer> s){
  s.push(1);
  s.push(2);
  s.push(3);
  s.push(4);
  while(s.size() > 0){
    System.out.println(s.pop());
  }
}
```

A. 1, 2, 3, 4
B. 1, 2, 4, 3
C. 4, 3, 2, 1
D. 3, 4, 2, 1